

Carlos Cubas Lorca

Proyecto de Fin de Grado

BICI RENT

Documentación Funcionalidad V1

Curso 2023-2024

Ciclo formativo Superior Desarrollo de Aplicaciones Web 2º DAW

Versión: v1

Fecha: 08/04/2024

HOJA DE CONTROL

Título	Carlos Cubas Lorca		
Entregable	Documentación del proyecto		
Nombre del Fichero	Documentación_Funcionalidad_V1.pdf		
Autor	Steria		
Versión/Edición	v01r00	Fecha Versión	08-04-2024
Aprobado por	Antonio Gabriel Gonzales Casado	Fecha Aprobación	19/06/2024
		Nº Total Páginas	011

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
V1	Modelo BD y URL Github	Antonio Gabriel Gonzales Casado	Desarrollo	08/04/2024

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias
Carlos Cubas	Estudiante	Desarrollo	1

ÍNDICE

1	Introducción	4
1.1	Descripción del Proyecto	4
1.2	Objetivo de la aplicación	4
1.3	Motivación para realizar el proyecto	4
2	Identificación de las necesidades del proyecto	5
2.1	Requisito Funcional RF01: Acciones para el Usuario Logueado (Cliente)	5
2.2	Requisito Funcional RF02: Acciones para el Usuario Logueado (Administrador)	6
2.3	Requisito Funcional RF03: Acciones para ambos tipos de Usuarios	7
3	Análisis y comparativa	8
3.1	Alternativas del Mercado en el Ámbito Nacional	8
3.2	Comparativa con nuestra Aplicación:	10
4	Justificación del proyecto:	10
4.1	Promoción de un estilo de vida sostenible	10
4.2	Optimización del uso de recursos:	11
4.3	Flexibilidad y conveniencia para los usuarios:	11
4.4	Fomento de la economía colaborativa:	11
5	Stack Tecnológico y Justificación del mismo	12
5.1	Frontend: React y Bootstrap:	12
5.2	Backend: Java Spring Boot:	12
5.3	Base de Datos: MYSQL:	12
5.4	Seguridad: Spring Security con JWT:	13
5.5	Documentación API: Swagger:	13
5.6	Pruebas: JUnit 5 y Mockito:	14
5.7	Herramientas de Gestión de Proyectos: Git, GitHub, Trello:	14
5.8	Validaciones: Implementación de validaciones:	15
5.9	Pruebas de Calidad: SonarLint:	15
5.10	Herramienta de Construcción:	15
5.11	Logging: LogBack + SLF4J:	15
6	Modelo de Base de datos	16
6.1	Casos de Uso:	16
6.2	Modelo Entidad Relación:	19
6.3	Descripción de las entidades del modelo conceptual	20
6.4	Descripción de los atributos de las tablas del modelo conceptual	22
6.5	Descripción de las relaciones del modelo conceptual	24
6.6	Modelo físico	26
7	GLOSARIO	27
8	BIBLIOGRAFÍA Y REFERENCIAS	28

1 Introducción

1.1 Descripción del Proyecto

El proyecto consiste en el desarrollo de una plataforma de alquiler de bicicletas, que ofrece a los usuarios la posibilidad de alquilar y publicar bicicletas de forma fácil y conveniente. La aplicación contará con un frontend desarrollado en React y Bootstrap para una experiencia de usuario moderna y receptiva. Por otro lado, el backend estará implementado en Java Spring Boot, proporcionando una arquitectura robusta y escalable, y la base de datos se gestionará con MySQL para almacenar y gestionar los datos de manera eficiente.

1.2 Objetivo de la aplicación

El objetivo principal de la aplicación es facilitar el acceso a bicicletas de calidad y promover su uso como medio de transporte sostenible. Al ofrecer una plataforma donde los usuarios puedan alquilar bicicletas directamente entre ellos, la aplicación busca fomentar un estilo de vida activo y respetuoso con el medio ambiente. Además, la aplicación pretende proporcionar una alternativa económica y práctica para aquellos que necesitan una bicicleta de forma temporal o ocasional.

1.3 Motivación para realizar el proyecto

El objetivo principal de la aplicación es facilitar el acceso a bicicletas de calidad y promover su uso como medio de transporte sostenible. Al ofrecer una plataforma donde los usuarios puedan alquilar bicicletas directamente entre ellos, la aplicación busca fomentar un estilo de vida activo y respetuoso con el medio ambiente. Además, la aplicación pretende proporcionar una alternativa económica y práctica para aquellos que necesitan una bicicleta de forma temporal o ocasional.

2 Identificación de las necesidades del proyecto

2.1 Requisito Funcional RF01: Acciones para el Usuario Logueado (Cliente)

2.1.1 Registro y Autenticación

Requisito Funcional	Descripción
RF01.1	Registrarse como nuevo usuario en la plataforma.
RF01.2	Iniciar sesión en la cuenta existente.

2.1.2 Exploración de Bicicletas

Requisito Funcional	Descripción
RF01.3	Explorar el catálogo de bicicletas disponibles para alquilar.
RF01.4	Filtrar y ordenar las bicicletas según sus preferencias (ubicación, tipo, precio, etc.).
RF01.5	Ver detalles completos de una bicicleta específica, incluyendo descripción, características y reseñas.

2.1.3 Reservas y Alquileres

Requisito Funcional	Descripción
RF01.6	Reservar una bicicleta seleccionada para una fecha y hora específicas.
RF01.7	Realizar el pago de alquileres a través de métodos seguros de pago en línea.

2.1.4 Perfil de Usuario de Usuario

Requisito Funcional	Descripción
RF01.8	Ver y editar su perfil de usuario, incluyendo información personal y detalles de contacto.
RF01.9	Ver el historial de alquileres anteriores, incluyendo detalles como fechas, duración y costos.
RF01.10	Gestionar sus publicaciones de bicicletas, incluyendo agregar, editar y eliminar publicaciones.

2.2 Requisito Funcional RF02: Acciones para el Usuario Logueado (Administrador)

2.2.1 Gestión del Sistema

Requisito Funcional	Descripción
RF02.1	Acceder a todas las funcionalidades de un usuario logueado estándar
RF02.2	Gestionar el inventario de bicicletas, incluyendo agregar, editar y eliminar bicicletas
RF02.3	Revisar y aprobar las publicaciones de bicicletas creadas por los usuarios.
RF02.4	Ver la lista de todos los usuarios registrados en la plataforma.

2.2.2 Gestión de Usuarios

Requisito Funcional	Descripción
RF02.5	Editar la información de los usuarios, incluyendo nombre de usuario, correo electrónico y detalles adicionales.
RF02.6	Eliminar cuentas de usuario según sea necesario para mantener la integridad de la plataforma.

2.2.3 Informes y Estadísticas

Requisito Funcional	Descripción
RF02.7	Generar informes sobre el uso de bicicletas, incluyendo estadísticas de alquileres y ganancias generadas.

2.3 Requisito Funcional RF03: Acciones para ambos tipos de Usuarios

2.3.1 Gestión del Sistema

Requisito Funcional	Descripción
RF03.1	Actualizar la información del perfil de usuario, incluyendo nombre de usuario, correo electrónico y contraseña.
RF03.2	Cerrar sesión de la cuenta actual para proteger la privacidad.

2.3.2 Gestión de Usuarios

Requisito Funcional	Descripción
RF03.3	Realizar búsquedas en el catálogo de bicicletas utilizando criterios específicos de búsqueda.
RF03.4	Contactar con otros usuarios a través de un sistema de mensajería interna para coordinar detalles de alquiler.

2.3.3 Informes y Estadísticas

Requisito Funcional	Descripción
RF03.5	Recibir notificaciones sobre el estado de los alquileres, como confirmaciones, recordatorios de devolución, etc.
RF03.6	Acceder al servicio de atención al cliente para resolver problemas o recibir asistencia técnica.

Estas acciones proporcionan una visión completa de las funcionalidades que los diferentes tipos de usuarios pueden realizar en la aplicación de alquiler de bicicletas, permitiendo una experiencia personalizada y satisfactoria para cada uno de ellos.

3 Análisis y comparativa

3.1 Alternativas del Mercado en el Ámbito Nacional

En el mercado español, existen varias aplicaciones populares de alquiler de bicicletas que ofrecen servicios similares a nuestra propuesta de aplicación, donde los usuarios pueden alquilar bicicletas directamente entre sí. A continuación, se presenta una comparativa entre nuestra aplicación y algunas de estas alternativas:

3.1.1 Wallapop y otras plataformas de compraventa entre particulares:

Ventajas:

- Plataformas bien establecidas y ampliamente utilizadas en España, con una gran base de usuarios.
- Permiten a los usuarios publicar anuncios de bicicletas disponibles para alquilar y comunicarse directamente para coordinar el alquiler.
- Ofrecen flexibilidad en términos de opciones de alquiler y precios, ya que los usuarios pueden negociar directamente entre sí.

Desventajas:

- No se especializan exclusivamente en el alquiler de bicicletas, lo que puede dificultar la búsqueda de opciones específicas.
- La calidad y disponibilidad de las bicicletas pueden variar ampliamente según el vendedor.
- Carecen de características específicas para la gestión de alquileres, como reservas, pagos seguros o seguimiento de tiempo de alquiler.

3.1.2 Amovens y otras plataformas de alquiler de vehículos entre particulares:**Ventajas:**

- Ofrecen un sistema de reservas y pagos seguro, proporcionando una experiencia más confiable para los usuarios.
- Amplia red de usuarios en España, con opciones de alquiler disponibles en múltiples ciudades.
- Permiten a los usuarios alquilar bicicletas directamente entre sí, sin intermediarios.

Desventajas:

- No se centran específicamente en el alquiler de bicicletas, lo que puede limitar la disponibilidad de opciones específicas.
- Las tarifas de alquiler pueden ser más altas debido a las comisiones de la plataforma.
- Pueden carecer de características avanzadas para la gestión de alquileres, como seguimiento de tiempo de alquiler o integración de pagos en línea.

3.2 Comparativa con nuestra Aplicación:

Ventajas de nuestra Aplicación:

- Especialización en el alquiler de bicicletas entre usuarios, lo que garantiza una experiencia centrada y específica para los usuarios que buscan alquilar bicicletas.
- Funcionalidades avanzadas para la gestión de alquileres, como reservas y seguimiento del tiempo de alquiler, que proporcionan una experiencia óptima para los usuarios.
- Interfaz intuitiva y fácil de usar, diseñada específicamente para facilitar la búsqueda, reserva y alquiler de bicicletas entre usuarios.

Desventajas potenciales de nuestra Aplicación:

- Puede requerir una base de usuarios inicial para alcanzar la misma escala que plataformas más establecidas como Wallapop o Amovens.
- Es posible que necesite una estrategia de marketing sólida para atraer y retener a los usuarios en un mercado competitivo de alquiler de bicicletas en España.
- La disponibilidad de bicicletas puede ser limitada inicialmente, especialmente en comparación con plataformas más amplias que incluyen múltiples categorías de productos.

4 Justificación del proyecto:

La justificación del proyecto se basa en la necesidad de ofrecer una solución innovadora y práctica para el alquiler de bicicletas, teniendo en cuenta las siguientes motivaciones:

4.1 Promoción de un estilo de vida sostenible

En la actualidad, existe una creciente conciencia sobre la importancia de adoptar medios de transporte más sostenibles para reducir la contaminación y mitigar el cambio climático. El fomento del uso de bicicletas como medio de transporte contribuye a este objetivo al reducir las emisiones de gases de efecto invernadero y promover hábitos de vida más saludables.

4.2 Optimización del uso de recursos:

Muchas personas poseen bicicletas que no utilizan con frecuencia debido a diversas razones, como la disponibilidad de transporte público o privado. Esta aplicación permite a los propietarios de bicicletas rentabilizar sus activos alquilados a otros usuarios interesados, optimizando así el uso de recursos y reduciendo el desperdicio.

4.3 Flexibilidad y conveniencia para los usuarios:

Al proporcionar una plataforma digital donde los usuarios pueden publicar y alquilar bicicletas entre sí, se ofrece una alternativa más conveniente y flexible en comparación con los servicios tradicionales de alquiler de bicicletas. Los usuarios tienen la libertad de elegir entre una variedad de opciones de bicicletas y horarios de alquiler que se adapten a sus necesidades individuales.

4.4 Fomento de la economía colaborativa:

La aplicación promueve la economía colaborativa al facilitar el intercambio de recursos entre los miembros de la comunidad. Permite a los usuarios generar ingresos adicionales alquilando sus bicicletas y ofrece a otros usuarios una alternativa económica y accesible para el transporte.

En resumen, la realización de esta aplicación web no solo responde a la creciente demanda de soluciones sostenibles y convenientes para la movilidad urbana, sino que también promueve valores de colaboración, eficiencia y responsabilidad ambiental en la sociedad.

5 Stack Tecnológico y Justificación del mismo

El uso del stack tecnológico se basa en la selección de herramientas y tecnologías que mejor se adapten a los requisitos del proyecto y que permitan desarrollar una aplicación robusta, eficiente y escalable. A continuación, se presenta la lista de tecnologías utilizadas para la construcción de la aplicación, junto con una justificación del por qué se han seleccionado:

5.1 Frontend: React y Bootstrap:

5.1.1 React:

React es una biblioteca de JavaScript que proporciona un enfoque eficiente y declarativo para construir interfaces de usuario interactivas. Su arquitectura basada en componentes facilita la reutilización del código y la creación de interfaces dinámicas y responsivas.

5.1.2 Bootstrap:

Bootstrap es un framework de CSS que ofrece un conjunto de herramientas y estilos predefinidos para el diseño de interfaces web modernas y adaptables. Su flexibilidad y compatibilidad con React facilitan la creación de interfaces consistentes y visualmente atractivas.

5.2 Backend: Java Spring Boot:

5.2.1 Java Spring Boot:

Spring Boot es un framework de desarrollo de aplicaciones Java que simplifica la configuración y el desarrollo de aplicaciones empresariales. Su amplia comunidad, su amplio ecosistema de bibliotecas y su soporte para principios de diseño como la inversión de control (IoC) y la inyección de dependencias hacen de Spring Boot una elección sólida para construir el backend de la aplicación.

5.3 Base de Datos: MYSQL:

5.3.1 MySQL:

MySQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado en aplicaciones web. Ofrece un rendimiento confiable, escalabilidad y una amplia compatibilidad con herramientas de desarrollo. Además, su integración con Spring Boot a través de JPA (Java Persistence API) simplifica la manipulación de datos en la aplicación.

5.4 Seguridad: Spring Security con JWT:

5.4.1 Spring Security:

Spring Security es un módulo de seguridad de Spring que proporciona autenticación y autorización robustas para aplicaciones Java. Su integración con Spring Boot permite implementar fácilmente características de seguridad como la autenticación basada en tokens y el control de acceso a recursos.

5.4.2 JWT (JSON Web Tokens):

JWT es un estándar abierto que define un formato compacto y autónomo para la transferencia segura de información entre partes como un objeto JSON. Se utiliza para la autenticación y autorización de usuarios en aplicaciones web modernas, proporcionando un mecanismo seguro y eficiente para la gestión de sesiones.

5.5 Documentación API: Swagger:

5.5.1 Swagger:

Swagger es una herramienta de código abierto que facilita la documentación, el diseño y la implementación de APIs RESTful. Permite a los desarrolladores describir, probar y visualizar la API de forma interactiva, lo que mejora la comprensión y la colaboración en el desarrollo de la aplicación.

5.6 Pruebas: JUnit 5 y Mockito:

5.6.1 JUnit 5:

JUnit 5 es un marco de pruebas unitarias para Java que proporciona un conjunto de herramientas para escribir y ejecutar pruebas de manera eficiente y efectiva. Su integración con Spring Boot facilita la realización de pruebas unitarias y de integración en el backend de la aplicación.

5.6.2 Mockito:

Mockito es un marco de pruebas para Java que permite la creación de objetos simulados (mocks) para probar el comportamiento de los componentes de la aplicación de manera aislada. Se utiliza en combinación con JUnit para realizar pruebas de unidades y de integración de forma fácil y flexible.

5.7 Herramientas de Gestión de Proyectos: Git, GitHub, Trello:

5.7.1 Git y GitHub:

Git es un sistema de control de versiones distribuido que permite el seguimiento de cambios en el código fuente de la aplicación. GitHub es una plataforma de alojamiento de código que proporciona funcionalidades adicionales como control de acceso, seguimiento de problemas y colaboración en proyectos.

5.7.2 Trello:

Trello es una herramienta de gestión de proyectos basada en tableros que permite organizar y seguir el progreso de las tareas en un entorno colaborativo. Se utiliza para planificar, asignar y realizar un seguimiento de las actividades del proyecto de manera efectiva.

5.8 Validaciones: Implementación de validaciones:

Se implementarán validaciones tanto en el frontend como en el backend para garantizar la integridad y la consistencia de los datos en la aplicación. Esto incluye la validación de formularios en el frontend mediante bibliotecas y la validación de datos en el backend mediante anotaciones de validación de Spring.

5.9 Pruebas de Calidad: SonarLint:

Se implementarán validaciones tanto en el frontend como en el backend para garantizar la integridad y la consistencia de los datos en la aplicación. Esto incluye la validación de formularios en el frontend mediante bibliotecas y la validación de datos en el backend mediante anotaciones de validación de Spring.

5.9.1 SonarLint:

SonarLint es una herramienta de análisis estático de código que ayuda a identificar y corregir problemas de calidad de código, vulnerabilidades de seguridad y malas prácticas de codificación. Se integra con IDEs como IntelliJ IDEA para proporcionar retroalimentación en tiempo real sobre la calidad del código.

5.10 Herramienta de Construcción:

5.10.1 Maven:

Maven es una herramienta de construcción y gestión de proyectos que automatiza el proceso de compilación, prueba y despliegue de aplicaciones Java. Proporciona un sistema de gestión de dependencias y una estructura de proyecto estandarizada que simplifica el desarrollo y la administración de proyectos.

5.11 Logging: LogBack + SLF4J:

5.11.1 SLF4J (Simple Logging Facade for Java):

SLF4J es una fachada de registro (logging facade) para Java que proporciona una capa de abstracción sobre diversos sistemas de registro de Java (como LogBack, Log4j, etc.). Permite a los desarrolladores escribir mensajes de registro en su código sin acoplar directamente la aplicación a una implementación específica de registro.

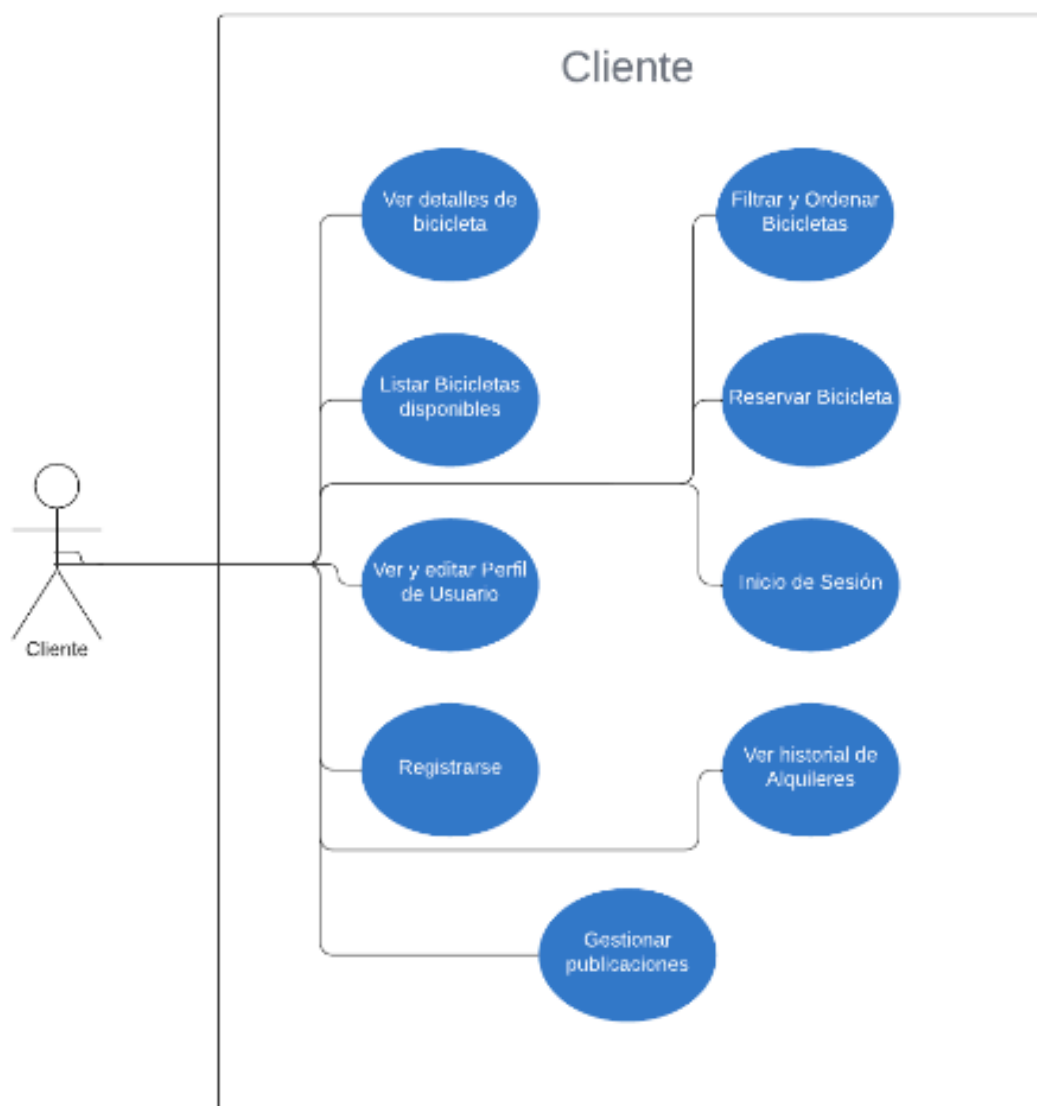
5.11.2 LogBack:

LogBack es una implementación de registro (logger implementation) compatible con SLF4J que ofrece funcionalidades avanzadas de registro y seguimiento de registros. Se utiliza en conjunto con SLF4J para proporcionar un mecanismo de registro flexible y configurable en la aplicación.

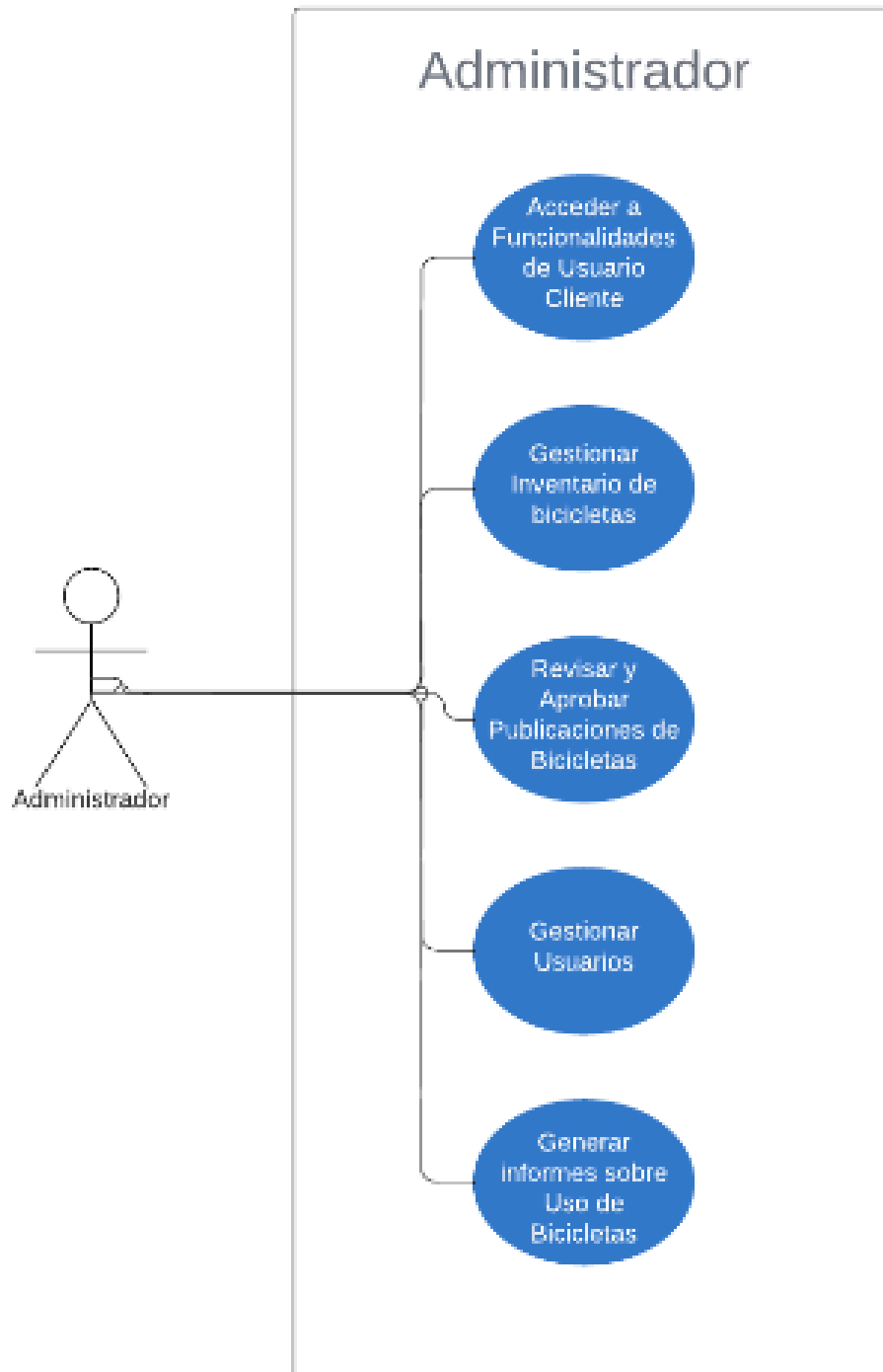
6 Modelo de Base de datos

6.1 Casos de Uso:

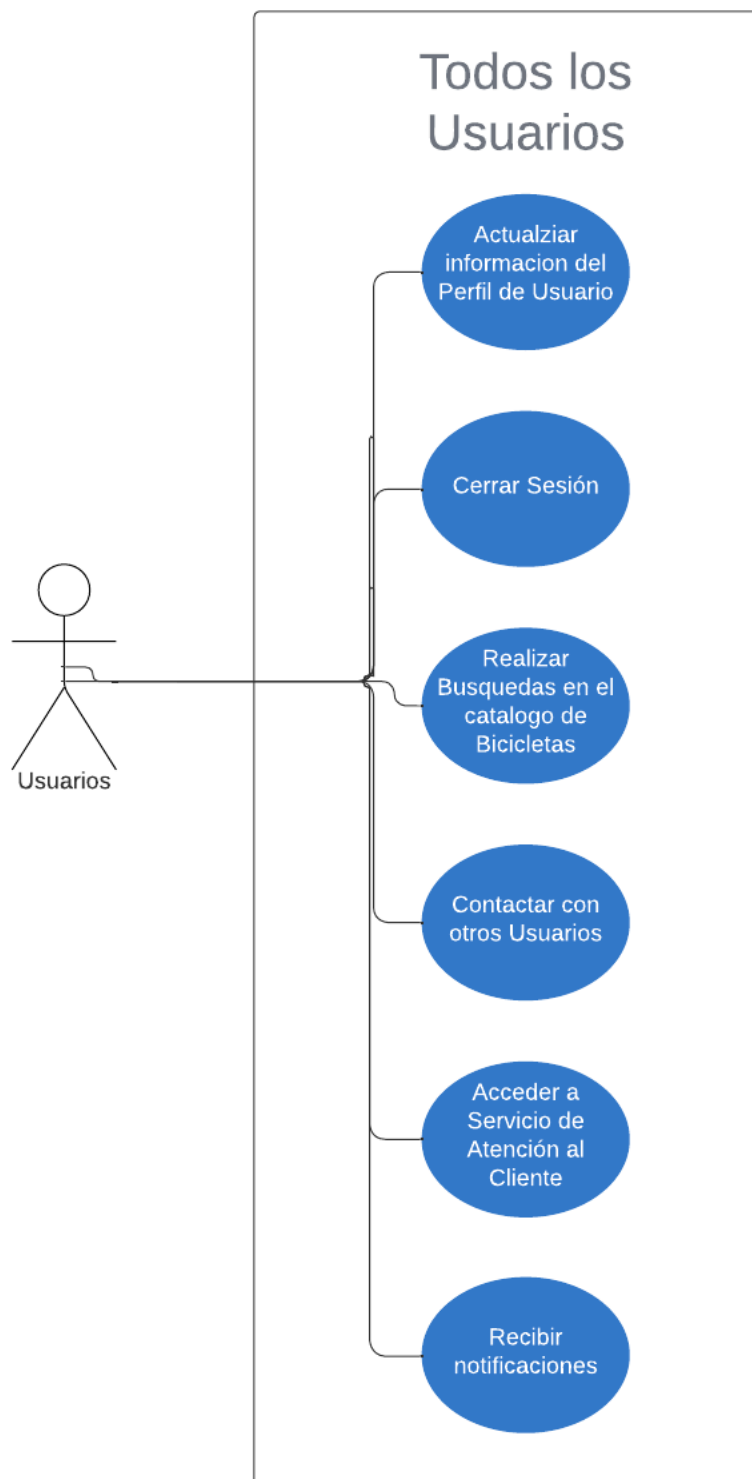
6.1.1 Funcionalidades del Usuario Logueado



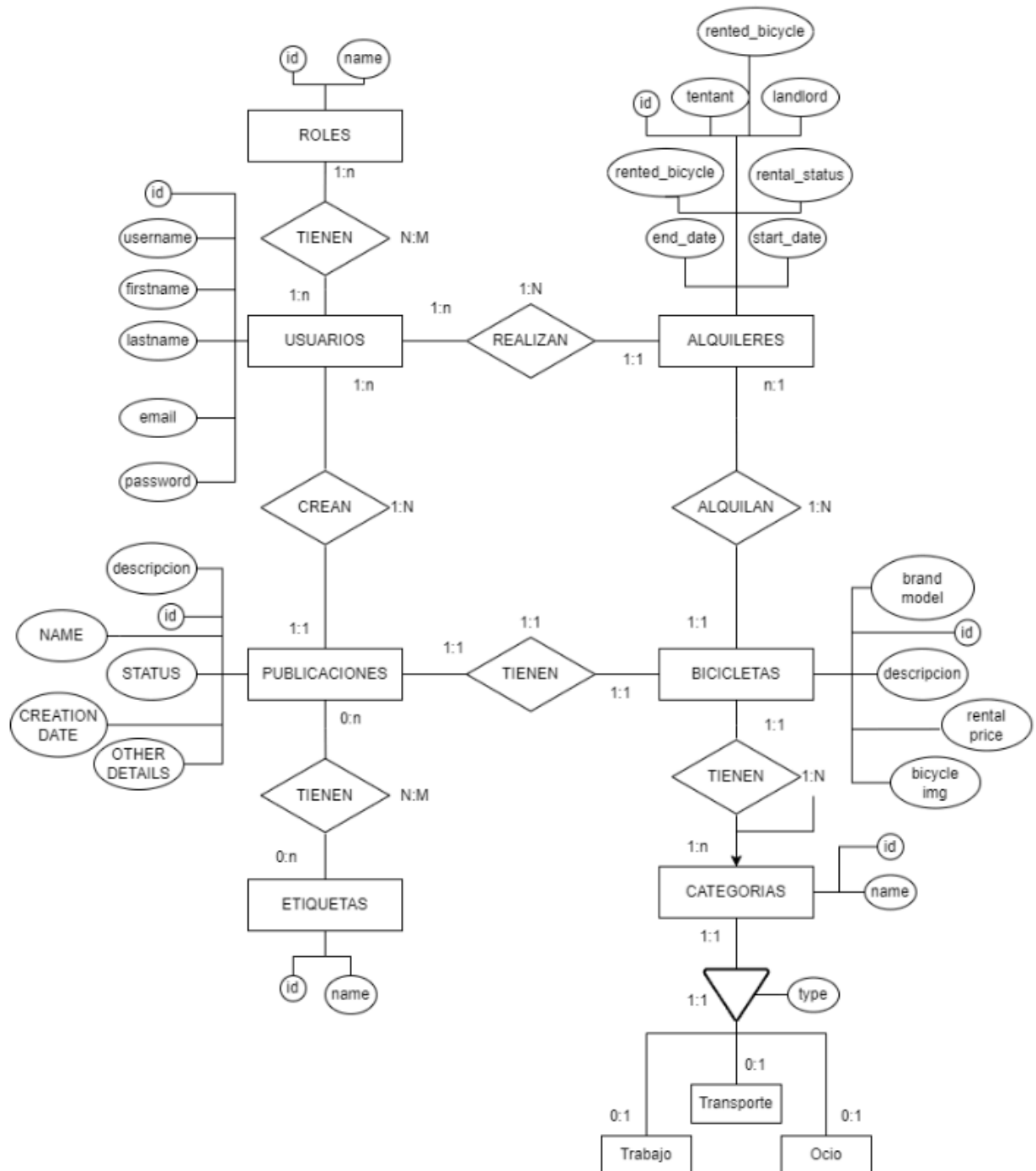
6.1.2 Funcionalidades del Usuario Administrador



6.1.3 Funcionalidades de todos los usuarios



6.2 Modelo Entidad Relación:



6.3 Descripción de las entidades del modelo conceptual

A continuación se describe las entidades en el modelo Entidad-Relación:

6.3.1 Entidad ROLES

Descripción: Esta tabla almacena los roles de los usuarios en el sistema, lo que permite asignar diferentes niveles de acceso y permisos a los usuarios según su función.

Utilidad: Sirve para controlar y gestionar los roles de los usuarios, como administradores, usuarios regulares, moderadores, etc., permitiendo establecer y limitar las acciones que pueden realizar en la plataforma.

6.3.2 Entidad USERS

Descripción: Almacena la información de los usuarios registrados en el sistema, incluyendo detalles como nombres, apellidos, direcciones de correo electrónico y contraseñas.

Utilidad: Esta tabla es fundamental para la gestión de usuarios, permitiendo el registro, inicio de sesión, recuperación de contraseñas y almacenamiento de información personal asociada a cada usuario.

6.3.3 Entidad USER_ROLES

Descripción: Es una tabla de relación entre usuarios y roles, que vincula a cada usuario con su respectivo rol en el sistema.

Utilidad: Permite asignar y gestionar los roles de los usuarios, controlando así su acceso y permisos dentro del sistema, lo que es crucial para mantener la seguridad y la integridad de la plataforma.

6.3.4 Entidad CATEGORIES

Descripción: Almacena las categorías de bicicletas disponibles en el sistema, clasificándolas según su tipo (Transporte, Trabajo, Ocio).

Utilidad: Facilita la organización y búsqueda de bicicletas por categoría, proporcionando a los usuarios una forma intuitiva de encontrar el tipo de bicicleta que están buscando.

6.3.5 Entidad TAGS

Descripción: Esta tabla almacena las etiquetas asociadas a las publicaciones de bicicletas, que ayudan a categorizar y clasificar las publicaciones de manera más detallada.

Utilidad: Permite una búsqueda más precisa y específica de las publicaciones de bicicletas, ya que los usuarios pueden filtrar las publicaciones por etiquetas relevantes, como "montaña", "urbana", "eléctrica", etc.

6.3.6 Entidad BICYCLES

Descripción: Almacena la información detallada de las bicicletas disponibles para alquilar, incluyendo datos como marca, modelo, descripción, precio de alquiler, etc.

Utilidad: Proporciona una lista completa de las bicicletas disponibles en el sistema, permitiendo a los usuarios ver y seleccionar la bicicleta que mejor se adapte a sus necesidades y preferencias.

6.3.7 Entidad RENTALS

Descripción: Almacena la información de los alquileres de bicicletas realizados por los usuarios, incluyendo detalles como el arrendador, arrendatario, fecha y hora de inicio y fin del alquiler, estado del alquiler, etc.

Utilidad: Registra y gestiona los alquileres de bicicletas en el sistema, permitiendo un seguimiento preciso de las transacciones de alquiler y facilitando la administración y la resolución de cualquier problema que pueda surgir durante el proceso de alquiler.

6.4 Descripción de los atributos de las tablas del modelo conceptual

A continuación se describen los campos de las entidades en el modelo Entidad-Relación:

6.4.1 Entidad ROLES

ROLE_ID: Identificador único del rol (clave primaria).

NAME: Nombre del rol, por ejemplo, "ROLE_USER" o "ROLE_ADMIN".

6.4.2 Entidad USERS

USER_ID: Identificador único del usuario (clave primaria).

USERNAME: Nombre de usuario único.

FIRSTNAME: Nombre del usuario.

LASTNAME: Apellido del usuario.

EMAIL: Dirección de correo electrónico del usuario.

PASSWORD: Contraseña del usuario.

6.4.3 Entidad USER_ROLES

USER_ID: Identificador único del usuario (clave foránea).

ROLE_ID: Identificador único del rol asignado al usuario (clave foránea).

6.4.4 Entidad CATEGORIES

CATEGORY_ID: Identificador único de la categoría (clave primaria).

CATEGORY_NAME: Nombre de la categoría.

CATEGORY_TYPE: Tipo de categoría, especificado como un ENUM con valores como "Transporte", "Trabajo" u "Ocio".

6.4.5 Entidad TAGS

TAG_ID: Identificador único de la etiqueta (clave primaria).

TAG_NAME: Nombre de la etiqueta que se utiliza para categorizar las publicaciones.

6.4.6 Entidad POSTS

OWNER_USER_ID: Identificador único del usuario que realiza la publicación (clave foránea).

CATEGORY_ID: Identificador único de la categoría de la publicación (clave foránea).

DESCRIPTION: Descripción detallada de la publicación.

POST_NAME: Nombre o título de la publicación.

POST_STATUS: Estado de la publicación, por ejemplo, "Activa" o "Inactiva".

CREATION_DATE: Fecha y hora en que se creó la publicación.

OTHER_DETAILS: Detalles adicionales relacionados con la publicación.

6.4.7 Entidad POSTS_TAGS

POST_ID: Identificador único de la publicación (clave foránea).

TAG_ID: Identificador único de la etiqueta asociada a la publicación (clave foránea).

6.4.8 Entidad BICYCLES

BICYCLE_ID: Identificador único de la bicicleta (clave primaria).

POST_ID: Identificador único de la publicación asociada a la bicicleta (clave foránea).

OWNER_USER_ID: Identificador único del usuario propietario de la bicicleta (clave foránea).

CATEGORY_ID: Identificador único de la categoría de la bicicleta (clave foránea).

BICYCLE_BRAND_MODEL: Marca y modelo de la bicicleta.

DESCRIPTION: Descripción detallada de la bicicleta.

RENTAL_PRICE: Precio de alquiler de la bicicleta.

BICYCLE_IMAGE: Ruta de la imagen de la bicicleta.

6.4.9 Entidad RENTALS

RENTAL_ID: Identificador único del alquiler (clave primaria).

LANDLORD_USER_ID: Identificador único del usuario arrendador (clave foránea).

TENANT_USER_ID: Identificador único del usuario inquilino (clave foránea).

RENTED_BICYCLE_ID: Identificador único de la bicicleta alquilada (clave foránea).

START_DATE_TIME: Fecha y hora de inicio del alquiler.

END_DATE_TIME: Fecha y hora de finalización del alquiler.

RENTAL_STATUS: Estado del alquiler, "En curso" o "Finalizado".

6.5 Descripción de las relaciones del modelo conceptual

A continuación se describen las relaciones mostradas en el modelo Entidad-Relación:

6.5.1 Relación entre BR_USERS y BR_ROLES

Descripción: Esta relación establece que un usuario puede tener uno o varios roles en el sistema.

Utilidad: Permite asignar diferentes niveles de acceso y permisos a los usuarios según su función o rol en la plataforma.

6.5.2 Relación entre BR_USERS y BR_USERS_ROLES:

Descripción: Es una relación de muchos a muchos que vincula los usuarios con los roles asignados.

Utilidad: Permite asignar múltiples roles a un usuario y gestionar los roles de manera flexible, adaptándose a las necesidades del sistema y los permisos requeridos por cada usuario.

6.5.3 Relación entre BR_POSTS y BR_USERS:

Descripción: Esta relación indica que cada publicación de bicicleta pertenece a un usuario que la ha creado.

Utilidad: Permite identificar al propietario de cada publicación y gestionar las publicaciones de bicicletas de manera individualizada.

6.5.4 Relación entre BR_POSTS y BR_CATEGORIES:

Descripción: Establece que cada publicación de bicicleta está asociada a una categoría específica de bicicletas.

Utilidad: Facilita la clasificación y organización de las publicaciones de bicicletas, permitiendo a los usuarios filtrar y buscar bicicletas por categoría.

6.5.5 Relación entre BR_POSTS y BR_TAGS:

Descripción: Es una relación de muchos a muchos que vincula las publicaciones de bicicletas con las etiquetas asociadas.

Utilidad: Permite etiquetar y categorizar las publicaciones de bicicletas de manera más detallada, lo que mejora la capacidad de búsqueda y filtrado para los usuarios.

6.5.6 Relación entre BR_BICYCLES y BR_POSTS:

Descripción: Indica que cada bicicleta está asociada a una publicación de bicicleta.

Utilidad: Permite relacionar cada bicicleta disponible para alquilar con la publicación correspondiente, proporcionando información detallada sobre cada bicicleta y su disponibilidad.

6.5.7 Relación entre BR_BICYCLES y BR_CATEGORIES:

Descripción: Establece que cada bicicleta pertenece a una categoría específica de bicicletas.

Utilidad: Facilita la organización y clasificación de las bicicletas disponibles para alquilar, permitiendo a los usuarios encontrar fácilmente el tipo de bicicleta que están buscando.

6.5.8 Relación entre BR_RENTALS y BR_USERS:

Descripción: Indica que tanto el arrendador como el arrendatario de una bicicleta en un alquiler son usuarios registrados en el sistema.

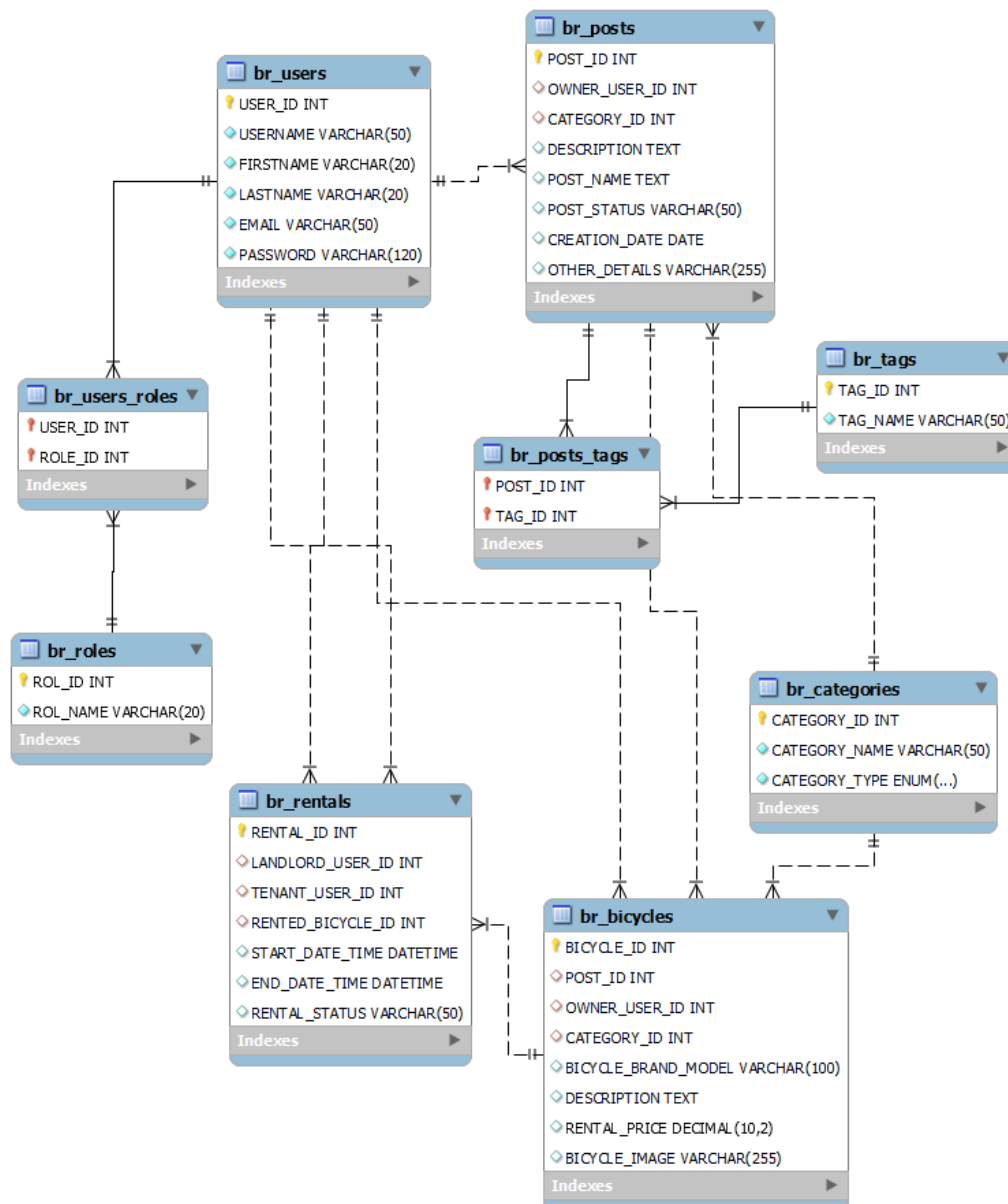
Utilidad: Permite identificar a los usuarios involucrados en cada transacción de alquiler y gestionar adecuadamente los registros de alquiler en el sistema.

6.5.9 Relación entre BR_RENTALS y BR_USERS:

Descripción: Indica que tanto el arrendador como el arrendatario de una bicicleta en un alquiler son usuarios registrados en el sistema.

Utilidad: Permite identificar a los usuarios involucrados en cada transacción de alquiler y gestionar adecuadamente los registros de alquiler en el sistema.

6.6 Modelo físico



7 GLOSARIO

Término	Descripción

8 BIBLIOGRAFÍA Y REFERENCIAS

Referencia	Título	Código
Ref. 1		